This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2779446, IEEE Transactions on Information Forensics and Security

Large-scale JPEG image steganalysis using hybrid deep-learning framework

Jishen Zeng, *Student Member, IEEE,* Shunquan Tan*, *Senior Member, IEEE,* Bin Li, *Senior Member, IEEE,* and Jiwu Huang, *Fellow, IEEE*

***Abstract*—Adoption of deep learning in image steganalysis is still in its initial stage. In this paper we propose a generic hybrid deep-learning framework for JPEG steganalysis incorporating the domain knowledge behind rich steganalytic models. Our proposed framework involves two main stages. The first stage is hand-crafted, corresponding to the convolution phase and the quantization & truncation phase of the rich models. The second stage is a compound deep neural network containing multiple deep subnets in which the model parameters are learned in the training procedure. We provided experimental evidences and theoretical reflections to argue that the introduction of threshold quantizers, though disable the gradient-descent-based learning of the bottom convolution phase, is indeed cost-effective. We have conducted extensive experiments on a large-scale dataset extracted from ImageNet. The primary dataset used in our experiments contains 500,000 cover images, while our largest dataset contains five million cover images. Our experiments show that the integration of quantization and truncation into deep-learning steganalyzers do boost the detection performance by a clear margin. Furthermore, we demonstrate that our framework is insensitive to JPEG blocking artifact alterations, and the learned model can be easily transferred to a different attacking target and even a different dataset. These properties are of critical importance in practical applications.**

***Index Terms*—hybrid deep-learning framework, CNN network, steganalysis, steganography.**

I. Introduction

IMAGE steganography can be divided into two main categories: spatial-domain and frequency-domain steganography. The latter focuses primarily on JPEG images due to their ubiquitous nature. Both categories in state-of-the-art algorithms adopt content-adaptive embedding schemes [1]. Most of these schemes use an additive distortion function defined as the sum of embedding costs of all changed elements. From early HUGO [2], to latest HILL [3] and MiPOD [4], the past few years witnessed the flourish of additive schemes in spatial domain. In JPEG domain, UED [5] and UERD [6] are two additive schemes with good security performance. UNIWARD proposed in [7] is an additive distortion function which can be applied for embedding both in spatial and JPEG domains. Its JPEG version, J-UNIWARD, achieves best performance [6], [7]. Research on non-additive distortion functions has made great progress in the spatial domain [8], [9]. However, analogous schemes have not yet been proposed in the JPEG domain. Although utilizing side information of a pre-cover image (raw or uncompressed) can improve the security of JPEG steganography [6], [7], [10], its applicability remains limited due to scarce availability of pre-cover images.

Most of modern universal steganalytic detectors use a rich model with tens of thousands of features [11]–[13] and an ensemble classifier [14]. In spatial domain, SRM [11] and its selection-channel-aware variants [12], [13] reign supreme. In JPEG domain, DCTR [15] feature set combines relatively low dimensionality and competitive performance, while PHARM [16] and GFR [17] exhibit better performance, although at the cost of higher dimensionality w.r.t. DCTR. SCA proposed in [18] is a selection-channel-aware variant of JPEG rich models targeted at content-adaptive JPEG steganography [1].

In recent years, with help of parallel computing accelerated by GPU (Graphics Processing Unit) and huge amounts of training data, deep learning frameworks have achieved overwhelming superiority over conventional approaches in many pattern recognition and machine learning problems [19]. Researchers in image steganalysis have also tried to investigate the potential of deep learning frameworks in this field. Tan et al. explored the application of stacked convolutional auto-encoders, a specific form of deep learning frameworks in image steganalysis [20]. Qian et al. proposed a steganalyzer based on CNN (Convolutional Neural Network) which achieving performance close to SRM [21], and demonstrated its transfer ability [22]. In [23], Pibre et al. revealed CNN based steganalyzers can achieve superior performance in the scenario that embedding key is reused for different stego images. Xu et al. constructed another CNN-based steganalyzer [24], [25] equipped with BN (Batch Normalization) layers [26]. Its performance slightly surpass SRM. In this paper the model proposed by Xu et al. in [24] is referred as Xu's model and is used for detection performance comparison. In [27], Sedighi and Fridrich implemented a specific CNN layer to imitate rich steganalytic model but failed to reached state-of-the-art performance. However, all of the above approaches [20]–[25], [27], focusing on spatial-domain steganalysis, are all evaluated on the BOSSBase (v1.01) dataset [28]. BOSSBase is arguably not representative of real-world steganalysis performance [29].

This work was supported in part by the NSFC (61772349, U1636202, 61402295, 61572329, 61702340), Guangdong NSF (2014A030313557), Shenzhen R&D Program (JCYJ20160328144421330). This work was also supported by Alibaba Group through Alibaba Innovative Research (AIR) Program. *(Corresponding author: Shunquan Tan.)*

S. Tan is with College of Computer Science and Software Engineering, Shenzhen University. J. Zeng, B. Li, and J. Huang are with College of Information Engineering, Shenzhen University.

All the members are with Shenzhen Key Laboratory of Media Security, Guangdong Province, 518060 China. (e-mail: tansq@szu.edu.cn).

[1]Throughout this paper, the acronyms used for the steganographic and steganalytic algorithms are taken from the original papers. The corresponding full names are omitted for brevity.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2779446, IEEE Transactions on Information Forensics and Security

With only 10,000 images, deep learning frameworks trained on BOSSBase are prone to overfitting. Furthermore, except our work which study the effect of fitting deep-learning steganalytic framework to a JPEG rich-model features extraction procedure [30], no prior works addressed the application of deep learning frameworks in JPEG steganalysis.

In this paper, we proposed a generic hybrid deep-learning framework for large-scale JPEG steganalysis. Our proposed framework combines the bottom hand-crafted convolutional kernels and threshold quantizers pairing with the upper compact deep-learning model. Experimental evidences and theoretical reflections are provided to show the rationale of our proposed framework. Furthermore, we have conducted extensive experiments on a large-scale dataset extracted from ImageNet [31] to demonstrate the capacity of our proposed generic framework under different scenarios.

The rest of the paper is organized as follows. In Sect. II, we describe the proposed hybrid deep-learning framework in detail, and provide experimental and theoretical testimonies to support its rationale. Results of experiments conducted on large-scale datasets are presented in Sect. III. Finally, we make a conclusion in Sect. IV.

II. Our proposed JPEG steganalytic framework

In this section, we firstly introduce the training procedure of CNN as preliminaries. Then we discuss the motivations and challenges related to the introduction of quantization and truncation in JPEG deep-learning steganalysis. Finally we describe our generic framework with experimental evidences and theoretical reflection to support our design.

A. Preliminaries

The principal part of CNN is a cascade of alternating convolutional layers, regulation layers (e.g. BN layers [26]) and pooling layers. On top of the principal part, there are usually multiple fully-connected layers. Please note that in CNN, only convolutional layers and fully-connected layers contain neuron units with learnable weights and biases [2]. Whether belongs to a convolutional layer or a fully-connected layer, each neuron unit receives inputs from a previous layer, performs a dot product with weights and optionally follows it with a nonlinear point-wise activation function. CNNs can be trained using backpropagation. For clarity, we omit those layers without learnable weights and biases, and denote the cascade of layers with learnable weights and biases in a given CNN as $[L_1, L_2, \cdots, L_n]$, where L_1 is the input layer and L_n is the output layer. $L_2, \cdots, L_{n-1}$ are the layers whose weights and biases are trained in backpropagation, namely convolutional layers and fully-connected layers. Let $a_i^{(l)}$ denote the activation (output) of unit i in layer L_l. For L_1, $a_i^{(1)}$ is the i-th input fed to the framework. $W_{ij}^{(l)}$ denotes the weight associated with unit i in L_l and unit j in L_{l+1}, while $b_j^{(l)}$ denotes the bias associated with unit j in L_{l+1}. The weighted sum of inputs to unit j in L_{l+1} is defined as:

$$z_j^{(l+1)} = \sum_i W_{ij}^{(l)} a_i^{(l)} + b_j^{(l)} \tag{1}$$

[2]The learnable parameters $\{\gamma, \beta\}$ for BN layers are omitted for brevity.

and $a_j^{(l+1)} = f(z_j^{(l+1)})$ where $f(\cdot)$ is the activation function. The set of all $W_{ij}^{(l)}$ and $b_j^{(l)}$ constitutes the parameterization of a neural network and is denoted as W and b, respectively. For a mini-batch of training features-label pairs $\{(x^{(1)}, y^{(1)}), \cdots, (x^{(m)}, y^{(m)})\}$, the goal of backpropagation is to minimize the overall cost function $J(W, b)$ with respect to W and b:

$$J(W, b) = \frac{1}{m} \sum_{h=1}^{m} J(W, b; x^{(h)}, y^{(h)}) + R(W) \tag{2}$$

where $R(W)$ is a regularization term which suppresses the magnitude of the weights, and $J(W, b; x^{(h)}, y^{(h)})$ is an error metric with respect to a single example $(x^{(h)}, y^{(h)})$.[3] For each training sample, the backpropagation algorithm firstly performs a feedforward pass and computes the activations for layers L_2, L_3 and so on up to the output layer L_n. For the j-th output unit in the output layer L_n, set the corresponding partial derivative of $J(W, b; x^{(h)}, y^{(h)})$ with respect to $z_j^{(n)}$:

$$\vartheta_j^{(n)} = \frac{\partial}{\partial a_j^{(n)}} J(W, b; x^{(h)}, y^{(h)}) f'(z_j^{(n)}) \tag{3}$$

Then in the backpropagation pass, partial derivatives are propagated from L_n back to the second last layer L_2. For the j-th neuron unit in layer L_l, set:

$$\vartheta_j^{(l)} = (\sum_k W_{jk}^{(l)} \vartheta_k^{(l+1)}) f'(z_j^{(l)}) \tag{4}$$

The partial derivatives with respect to $W_{ij}^{(l)}$ and $b_j^{(l)}$, $l = n-1, n-2, \cdots, 1$ are calculated as:

$$\begin{cases} \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b; x^{(h)}, y^{(h)}) = a_i^{(l)} \vartheta_j^{(l+1)} \ , \\ \frac{\partial}{\partial b_j^{(l)}} J(W, b; x^{(h)}, y^{(h)}) = \vartheta_j^{(l+1)} \ , \end{cases} \tag{5}$$

Gradient descent is used to find the optimal W and b. In the optimization procedure, it updates W and b according to steps proportional to the negative of the average of m gradients each of which is the vector whose components are the partial derivatives in (5) [32].

B. The introduction of quantization and truncation in deep-learning based steganalysis

State-of-the-art rich models for JPEG steganalysis [15]–[18] take decompressed (non-rounded and non-truncated) JPEG images as input. The feature extraction procedure of JPEG rich models can be divided into three phases:

- *Convolution*: The target image is convolved with a set of kernels to generate diverse noise residuals. The purpose of this phase is to suppress the image contents as well as boost SNR (Signal-to-Noise Ratio).
- *Quantization and truncation* (Q&T): Different quantized and truncated versions of each residual are calculated to further improve diversity of resulting features, as well as reduce the computational complexity.
- *Aggregation*: The values in noise residuals are aggregated to further reduce feature dimensionality.

[3]There are various forms of $J(W, b; x^{(h)}, y^{(h)})$ and $R(W)$, and their definitions are omitted here, since irrelevant to the subject of this paper.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2779446, IEEE Transactions on Information Forensics and Security

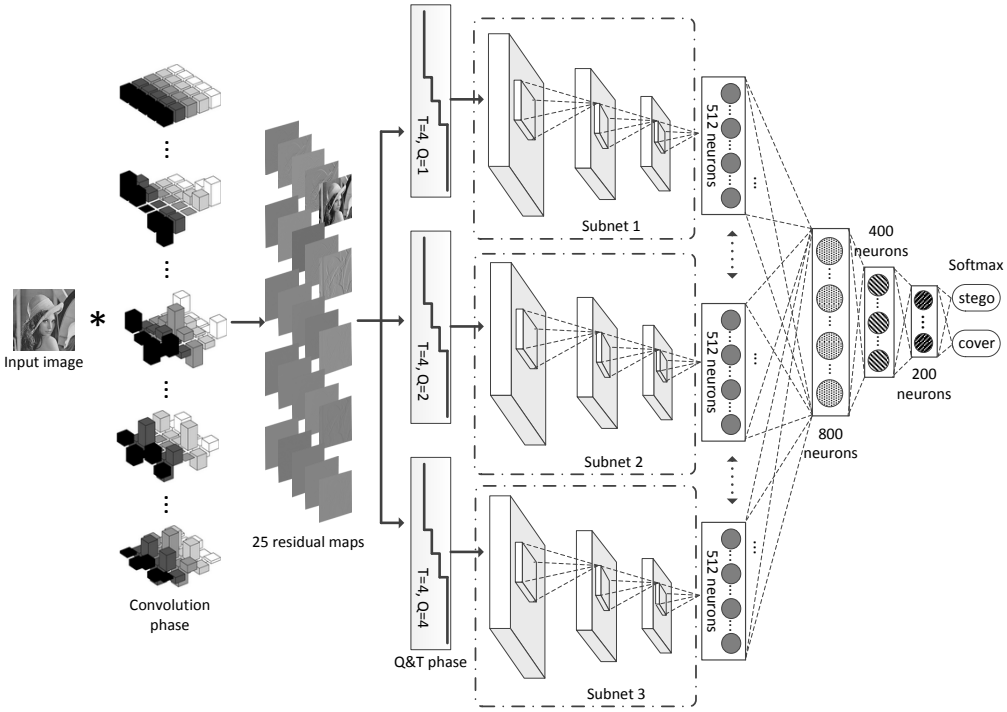


Fig. 1. Conceptual architecture of one implementation of our proposed hybrid deep-learning framework with twenty-five 5×5 DCT basis patterns and three Q&T combinations.

Take DCTR [15] for example. Given a $M \times N$ JPEG image, it is firstly decompressed to the corresponding spatial-domain version $\mathbf{X} \in \mathbb{R}^{M \times N}$. Sixty-four 8×8 DCT basis patterns are defined as $\mathbf{B}^{(k,l)} = (B^{(k,l)}_{mn}), 0 \le k, l \le 7, 0 \le m, n \le 7$:

$$B^{(k,l)}_{mn} = \frac{w_k w_l}{4} cos \frac{\pi k(2m+1)}{16} cos \frac{\pi l(2n+1)}{16}, \tag{6}$$

where $w_0 = \frac{1}{\sqrt{2}}$, $w_k = 1$ for $k > 0$. $\mathbf{X}$ is convolved with $\mathbf{B}^{(k,l)}$ to generate 64 noise residuals $\mathbf{U}^{(k,l)}, 0 \le k, l \le 7$:

$$\mathbf{U}^{(k,l)} = \mathbf{X} * \mathbf{B}^{(k,l)}, \tag{7}$$

Then the elements in each $\mathbf{U}^{(k,l)}$ are quantized with quantization step q and truncated to a threshold T. The DCTR features are constructed based on certain aggregation operation that collect specific first-order statistics of the absolute values of the quantized and truncated elements in each $\mathbf{U}^{(k,l)}$.

In [20], we pointed out that in general the above structure of rich models resembles CNN. Quantization and truncation has become an indispensable part of rich steganalytic models [11]–[13], [15]–[18]. However, as far as we know, there still has been no published works regarding to the integration of quantization and truncation into deep-learning steganalyzers.

In this paper, we would like to utilize the domain knowledge behind rich models, especially the specific kernel matrices in the convolutional phase and the Q&T phase. But, The introduction of quantization and truncation, namely the Q&T phase on top of the bottom convolution phase, is a double-edged sword. It cannot be put in the pipeline of gradient-descent-based learning. The Q&T phase takes noise residuals generated by convolution phase as input, and can be modeled as:

$$a^{(2)}_j = f(z^{(2)}_j) = \begin{cases} \min([z^{(2)}_j/q], T) & \text{if } z^{(2)}_j >= 0 \\ \max([z^{(2)}_j/q], -T) & \text{if } z^{(2)}_j < 0 \end{cases} \tag{8}$$

where $z^{(2)}_j$ is an element of a given noise residual generated by the bottom convolution phase, $a^{(2)}_j$ is the corresponding activation output, q is the quantization step, $[\cdot]$ denotes the rounding operation, and T is a predefined threshold. It is obvious that $f'(z^{(2)}_i)$ is zero along the entire domain of $z^{(2)}_j$ except the set of points $\{(-T+0.5)q, (-T+1.5)q, \cdots, (T-1.5)q, (T-0.5)q\}$ where it is infinite. Therefore (8) cannot be put in the pipeline of gradient descent, since the derivative it passes on in backpropagation will vanish. More specifically, the derivative does not exist if $z^{(2)}_j$ is located at one of the points in the set $\{(-T+0.5)q, (-T+1.5)q, \cdots, (T-1.5)q, (T-0.5)q\}$, otherwise the derivative is equal to zero. The corresponding gradient saturates if the partial derivative it passes on approaches to zero, and is nullified if there is no derivative.

Incompatibility between Q&T phase and gradient-descent-based learning presents a dilemma in the design of deep-learning steganalytic framework. The introduction of Q&T phase implies that gradient descent cannot be back propagated to the bottom convolution phase without the usage of some unconventional bypass trick. The generic hybrid deep-learning framework for JPEG steganalysis proposed in this paper is intended to provide a solution to this dilemma.

C. Our proposed hybrid deep-learning framework

Our proposed generic framework is composed of two stages. The first stage takes decompressed (non-rounded and non-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2779446, IEEE Transactions on Information Forensics and Security

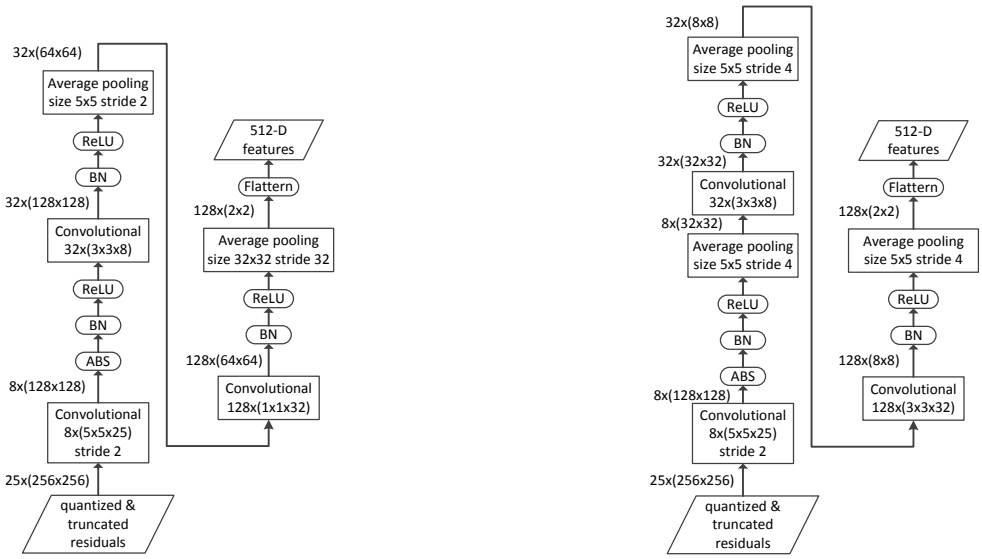


Fig. 2. Two types of subnet configurations. (a) *Type1* subnet. (b) *Type2* subnet. In the two figures "ABS" denotes the activation layer which outputs absolute values of the corresponding inputs, "BN" denotes the batch normalization layer, and "ReLU" denotes the layer with rectified-linear-unit activation functions.

truncated) JPEG images as input, and corresponds to the convolution phase and the Q&T phase of rich models. The proposed generic framework can be implemented in different way. The conceptual architecture of one implementation with twenty-five 5×5 DCT basis patterns and three Q&T combinations is illustrated in Fig. 1. In this implementation, the first stage incorporated the first two phases of DCTR [15]. All model parameters in this stage are hand-crafted and gradient-descent-based learning is disabled. What makes this stage different from DCTR is that DCTR uses sixty-four 8×8 DCT basis patterns and only one Q&T combination, while our proposed approach contains twenty-five 5×5 DCT basis patterns which are defined as $\mathbf{B}^{(k,l)} = (B_{mn}^{(k,l)}), 0 \leq k, l \leq 5, 0 \leq m, n \leq 5$:

$$B_{mn}^{(k,l)} = \frac{w_k w_l}{5} cos\frac{\pi k(2m+1)}{10} cos\frac{\pi l(2n+1)}{10}, \quad w_0 = 1,\ w_k = \sqrt{2} \text{ for } k > 0. \quad (9)$$

and three Q&T combinations, namely $(T = 4, Q = 1)$, $(T = 4, Q = 2)$ and $(T = 4, Q = 4)$. Given an input image, the convolution phase outputs twenty-five residual maps. The residual maps pass through the Q&T phase. Three different groups of quantized and truncated residual maps are generated. They constitute the input of the second stage. The intention behind the design of the first stage of our proposed framework is that we would like to utilize the domain knowledge behind rich models, especially the specific kernel matrices in the convolutional phase and the Q&T phase. We agree with the concepts in rich models [11]: model diversity is crucial to the performance of steganalytic detectors. The model diversity of our proposed framework is represented in twenty-five DCT basis patterns in the hand-crafted convolutional layer and the three Q&T combinations that followed. There are total $25 \times 3 = 75$ sub-models in our proposed framework.

The second stage is a compound deep CNN network in which the model parameters are learned in the training procedure. The bottom of the second stage is composed of three independent subnets with identical structure. Each subnet corresponds to one group of quantized and truncated residual maps. They take the residual maps as input and generate three feature vectors. As shown in Fig. 2, within this implementation, two types of subnet configurations are adopted. Both of them contain three convolutional layers and output a 512-D (512 dimensional) feature vector. *Type1* subnet (Fig. 2(a)) adopts 1×1 convolutional kernels in the top-most convolutional layer and uses a single average pooling layer with large 32×32 pooling windows at the end, as suggested in Xu's model [24]. However, deviated from the recipe suggested in Xu's model [24] that using TanH (Hyperbolic Tangent) activation function in the lower part, we always use ReLU (Rectified Linear Unit) activation function in *Type1* subnet. *Type2* subnet (Fig. 2(b)) is a traditional CNN configuration. Compared with *Type1* subnet, it adopts progressive pooling layers and uses 3×3 convolutional kernels in the top-most convolutional layer. Due to the progressing pooling layers, *Type2* subnet is a relative GPU memory-efficient model. The GPU memory requirement of *Type2* subnet is only one-seventh of that of *Type1* subnet. Both configurations have in common are the BN layers which follow every convolutional layer.

In this implementation, three 512-D feature vectors output by the bottom subnets are concatenated together to generate a single 1536-D feature vector. The feature vector is subsequently fed into a four-layer fully-connected neural network which makes the final prediction. The successive layers of the fully-connected network contain 800, 400, 200, and 2 neurons, respectively. ReLU activation functions are used in all three hidden layers. The final layer contains two neurons which denote "stego" prediction and "cover" prediction, respectively. Softmax function is used to output predicted probabilities.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2779446, IEEE Transactions on Information Forensics and Security

Recent researches on deep-learning revealed that ensemble prediction with independently trained deep-learning models can improve the performance [33]. In [25], Xu et al. also demonstrated the potential of ensemble prediction in deep-learning based steganalysis. Therefore, when compared to state of the art in Sect. III-C, we also introduce model ensemble in the final prediction in order to further promote the detection performance. Different from the approaches in [25], we adopt a simple ensemble strategy, like the one used in [33]. Five versions of our proposed deep-learning models are independently trained with the same learning setting and training dataset. They differ only in initial weights of the learnable stage. When testing, the decision of the five models are combined with majority voting.

There is significant difference between our proposed framework and other existing deep-learning steganalyzers [20]–[25], [27]. Firstly, we explicitly introduce the Q&T phase used in rich models into our proposed deep-learning steganalytic framework, which have never been seen in previous works. Secondly, we adopt an array of dozens of hand-crafted convolutional kernels in the bottom layer of our proposed framework, instead of an image pre-processing layer with a single high-pass filter used in previous works. And finally, there are three parallel CNN subnets with identical structure in the central portion of our proposed framework, which also have never been seen in previous works.

Our large-scale experiments reported in the following Sect. III demonstrated that the introduction of Q&T phase do bring substantial detection performance improvement. The performance improvement is not only due to the model diversity brought by different Q&T combinations (as shown in Sect. III-B). The discretization brought by quantization and truncation itself also has an obvious impact on the detection performance. We report the following experimental evidences to support our argument. The experiments were conducted on basic500K with setups shown in Sect. III-A. J-UNIWARD stego images with 0.4bpnzAC (bits per non-zero cover AC DCT coefficient) were included in the experiments. In the experiments our proposed framework was equipped with *Type1* subnet. A corresponding model was trained and tested independently for each configuration combination. We tested the trained model every $10,000$ iterations, and reported the best testing accuracy in 20×10^4 iterations. No ensemble prediction was involved in this experiment, as in Sect. III-B. The basic evidences are listed as follows:

- The detection accuracy of Xu's model [24], which is without Q&T phase, is merely 54.7%.
- The detection accuracy of our proposed framework as illustrated in Fig. 1 is 74.5%.
- The detection accuracy of our proposed framework without Q&T phase is 61.5%.
- The detection accuracy of our proposed framework without quantization step in the Q&T phase, is 57.6%, even worse than the above one without the entire Q&T phase.
- The detection accuracy of our proposed framework without truncation step in the Q&T phase, is 65.4%.

From the above experimental evidences we can clearly see that both quantization and truncation effectively improve the detection performance.

As mentioned in the last section, the introduction of Q&T phase implies that gradient descent cannot be back propagated to the bottom convolution phase. However, we still can back-propagate a fixed fake tiny derivative d to the bottom convolution phase. [4] However, our extensive experiments show that such a fake derivative just leads to serious performance degradation. For example, using a Q&T phase with a fixed fake derivative d, the detection accuracy of our proposed framework as illustrated in Fig. 1 is merely 60.5% when $d = 0.01$, and 56.8% when $d = 0.001$. Therefore, at present no compromise solution to the incompatibility can be found.

But, does gradient-descent optimization of the bottom convolution phase really matter? The following two experimental evidences reveal that gradient-descent optimization of the bottom convolution phase cannot improve the detection performance:

- The detection accuracy of Xu's model with a learnable bottom convolutional kernel, which is initialized as the high-pass filter used in [24], is 54.6%. Its performance is slightly worse than the one with fixed high-pass filter.
- The detection accuracy of our proposed framework without Q&T phase is 61.3%, under the condition that gradient-descent- based learning is enabled for the bottom convolution phase. Its performance is also slightly worse than the one with fixed DCT basis patterns.

Recently in a similar field, image forensics, Bayar et al. proposed a convolutional-layer regularizer which was claimed can be used to suppress the content of an image [34]. However, we observed that regularizing the bottom convolutional kernels using the approach in [34] did not lead to positive changes in the above two experimental evidences:

- The detection accuracy of Xu's model is still 54.6%.
- The detection accuracy of our proposed framework without Q&T phase is 61.2%, slightly worse than the prior one.

All of the above experimental evidences reveal that at least in the field of JPEG steganalysis, it is extraordinary difficult for an existing deep-learning steganalytic framework to benefit from gradient-descent optimization of the bottom convolution phase, under the premise that the kernels in the bottom convolution phase have already possessed the same parameters as those used in rich models. We attribute this difficulty to the contradiction between the design philosophy (or domain knowledge) of the kernels in rich models and the gradient descent algorithm used in deep-learning frameworks. The long and widely accepted philosophy behind rich steganalytic models is that high-pass kernels should be designed to extract the noise component (noise residual) of images rather than their content [11]. However, as shown in the theoretical reflection in Appendix A, for a deep-learning framework, we argue that the optimization of the bottom convolutional kernels in favor of

[4] In practice, fake partial derivative can be back propagated to bottom layers when the actual partial derivative vanishes. For example, this trick is used in the Caffe implementation of ReLU layer (https://github.com/BVLC/caffe/blob/master/src/caffe/layers/relu_layer.cpp).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2779446, IEEE Transactions on Information Forensics and Security

the extraction of stego noises is hard to achieve with gradient descent.

The experimental demonstration in this section indicates that the introduction of Q&T phase do bring substantial detection performance improvement. Certainly, the introduction of Q&T phase is with negative side effect: it blocks the back-propagated gradients. But the theoretical reflection in Appendix A shows that such negative side effect can be ignored, since even not cut off by Q&T phase, the back-propagated gradients is still hard to properly guide the optimization of the bottom convolutional layer, as long as its optimization goal is to benefit the extraction of stego noises. In fact, the authors believe that we cannot directly draw on the design philosophy of rich models to understand the underlying mechanism of deep-learning steganalytic framework. Deep-learning frameworks are trained and optimized as a whole. It may be not suitable to isolate one part of a given deep-learning framework (e.g. the learnable bottom convolutional layer) and force it to comply with existing design philosophy. Therefore our proposed hybrid deep-learning framework for JPEG steganalysis is designed to be composed of two stages. The bottom hand-crafted stage, which contains the convolution phase and the Q&T phase incorporated from rich models and complied with its design philosophy, is not involved in gradient-descent-based optimization. The second stage is a compound deep CNN network which does not need to comply with the design philosophy of rich models, and is free to be optimized using backpropagation as a whole.

III. Experimental results

A. Experiment setups

We adopted ImageNet [31], a large-scale image dataset containing more than fourteen million JPEG images, to evaluate the steganalytic performance of our proposed hybrid deep-learning framework. All of the experiments were conducted on a GPU cluster with eight NVIDIA® Tesla® K80 dual-GPU cards. Independent models are trained and tested in parallel, each of which is assigned one GPU. By considering the computation capacity, we restricted the size of the target images to 256×256. We randomly selected 50 thousand, 500 thousand and 5,000 thousand (namely 5 million) JPEG images with size larger than 256×256 from ImageNet. Their left-top 256×256 regions were cropped, converted to grayscale and then re-compressed as JPEG with quality factor 75.[5] The resulting images constituted the following three basic cover image datasets:

- basic50K: The small-scale dataset used in our experiments. By comparing the detection performance of our proposed framework on basic50K and basic500K (see below), we can highlight the superiority of our proposed framework in large-scale dataset.
- basic500K: The major dataset for most all of our experiments, including the verification experiments to determine hyper-parameters of our proposed framework.
- basic5000K: The largest-scale dataset used in our experiments. Due to the limitation of computation capacity, we only conducted the experiments on stego images with 0.4 bpnzAC.

Our implementation was based on the publicly available Caffe toolbox [35] with our implemented hand-crafted convolutional layer (with 5×5 DCT basis patterns) and Q&T layer according to (8). Our proposed models were trained using mini-batch stochastic gradient descent with "step" learning rate starting from 0.001 (stepsize: 5000; weight_decay: 0.0005; gamma: 0.9) and a momentum fixed to 0.9. The batch size in the training procedure was 64 and the maximum number of iterations was set to 20×10^4. In each experiment, we tested the trained model in the corresponding standalone testing set every $10,000$ iterations, and reported the best testing accuracy in 20×10^4 iterations. Please note that as later shown in Fig. 4, when trained on a large-scale dataset such as basic500K, our proposed framework exhibited good convergence and stability after less than 5×10^4 iterations. Therefore validation set was omitted for the sake of resources saving. The source code and auxiliary materials are available for download from GitHub [6].

J-UNIWARD [7], UERD [6] and UED [5], the three state-of-the-art JPEG steganographic schemes, were our attacking targets in the experiments. The default parameters of the three steganographic schemes were adopted in our experiments. 50% cover images were randomly selected from basic50K, basic500K, and basic5000K, respectively. They constituted the training set along with their corresponding stego images. The rest 50% cover-stego pairs in the dataset were for testing. We further guaranteed that the cover images included in an arbitrary training set of the three datasets would not appear in any of the three testing sets.

B. Impact of the framework architecture on the performance

In Tab. I, we compare the effect of different Q&T combinations, different hand-crafted convolutional kernels, and the presence of BN layers. The experiment was conducted on basic500K. A corresponding model is trained and tested independently for each configuration combination. No ensemble prediction is involved in this experiment. We can see that under the same conditions, DCT basis patterns (including the 8×8 DCTR kernels [15]) always perform better than PHARM kernels [16]. The experimental results support our choice of DCT basis patterns. 5×5 DCT basis patterns can achieve significant performance improvement compared to 3×3 DCT basis patterns. However, the performance of the more complex 8×8 DCTR kernels is not even as good as the 3×3 DCT basis patterns, which indicates that increasing the size of the convolutional kernels is not always beneficial at the cost

[5]The original quality factors of ImageNet images are diverse. Out of 10 million ImageNet images with size larger than 256×256, there are more than 1.5 million images whose quality factors cannot be detected by ImageMagick utility "identify", and roughly 8.3 million images with diverse quality factors which are larger than 75. We uniformly converted the quality factors of the selected images to 75 due to the following two reasons: Firstly, all the reported experiments of previous works, including DCTR, PHARM, GFR, and SCA, are conducted on images with quality factor 75 and 95. And secondly, if the target quality factor is set to 95, then for a majority of the selected images, we need to elevate their quality factors which may introduce exploitable artifacts.

[6]https://github.com/tansq/hybrid_deep_learning_framework_for_jpeg_steganalysis

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2779446, IEEE Transactions on Information Forensics and Security

of increasing model complexity. The performance of GFR kernels [17] is slightly better than 5×5 DCT basis patterns. However, with as many as two hundred and fifty-six output residual maps, GFR kernels are too resource consuming to be included in our proposed framework. Different Q&T combinations also affect the performance of our proposed framework. Combinations with three different quantization steps and the same threshold are of relatively cost-effective. BN layers in the subnets are crucial, especially the first one at the bottom of the subnets. Therefore, based on the described results, we adopt twenty-five 5×5 DCT basis patterns, $T = 4, Q = [1, 2, 4]$ and subnet configurations with a BN layer following every convolutional layer in our final proposed framework.

TABLE I

EFFECT OF DIFFERENT Q&T COMBINATIONS, DIFFERENT HAND-CRAFTED CONVOLUTIONAL KERNELS, AND THE PRESENCE OF BN LAYERS. ONLY J-UNIWARD STEGO IMAGES WITH 0.4BPNZAC WERE INCLUDED IN THE EXPERIMENTS. THE BEST RESULTS IN EVERY SUB-TABLE ARE UNDERLINED. THOSE HYPER-PARAMETERS ADOPTED IN OUR PROPOSED FRAMEWORK ARE MARKED IN BOLD.[a]

Threshold & Quantization Steps	BN Layers		
	With BNs	Without BN1	Without BNs
Nine 3×3 DCT basis patterns			
(4,1), (4,1.5), (4,2)	73.1%	70.6%	50.1%
(4,2), (4,2), (4,2)	72.8%	70.1%	50.0%
(4,1), (4,2), (4,4)	<u>73.2%</u>	71.0%	50.1%
(2,1), (4,2), (6,4)	71.2%	68.5%	50.0%
(6,1), (4,2), (2,4)	70.6%	67.8%	50.0%
Twenty-five 5×5 DCT basis patterns			
(4,1), (4,1.5), (4,2)	74.3%	72.4%	50.1%
(4,2), (4,2), (4,2)	74.1%	72.4%	50.1%
(4,1)	70.8%	69.4%	50.1%
(4,1), (4,2)	72.5%	70.2%	50.1%
(4,1), (4,2), (4,4)	<u>74.5%</u>	72.5%	50.1%
(2,1), (4,2), (6,4)	73.6%	72.0%	50.1%
(6,1), (4,2), (2,4)	72.6%	71.7%	50.0%
Sixty-four 8×8 DCTR kernels [15]			
(4,1), (4,1.5), (4,2)	72.5%	71.4%	50.0%
(4,2), (4,2), (4,2)	72.7%	71.2%	50.1%
(4,1), (4,2), (4,4)	<u>72.9%</u>	71.2%	50.1%
(2,1), (4,2), (6,4)	71.9%	70.2%	50.0%
(6,1), (4,2), (2,4)	71.5%	70.1%	50.1%
Thirty 5×5 PHARM kernels [16]			
(4,1), (4,1.5), (4,2)	72.0%	70.8%	50.1%
(4,2), (4,2), (4,2)	70.6%	68.8%	50.0%
(4,1), (4,2), (4,4)	<u>72.1%</u>	70.8%	50.1%
(2,1), (4,2), (6,4)	70.3%	68.6%	50.0%
(6,1), (4,2), (2,4)	70.2%	68.7%	50.0%
Two hundred and fifty-six 8×8 GFR kernels [17]			
(4,1), (4,1.5), (4,2)	74.1%	72.5%	50.1%
(4,2), (4,2), (4,2)	74.0%	72.6%	50.1%
(4,1), (4,2), (4,4)	<u>74.6%</u>	72.5%	50.0%
(2,1), (4,2), (6,4)	74.1%	72.4%	50.0%
(6,1), (4,2), (2,4)	72.3%	71.5%	50.0%

[a]Logograms are used in expressing Q&T combinations. For example, (4,1) denotes ($T = 4, Q = 1$).

C. Comparison to state of the art

In Fig. 3, we compare the performance of our proposed framework and other steganalytic models in the literature. Please note that for a fair comparison, Xu's model [24] is also fed with decompressed (non-rounded and non-truncated) images, and is trained with the same learning protocol as that for our own model [7]. From Fig. 3 we can see that our proposed framework can obtain significant performance improvement compared with DCTR [15], GFR [16], and even recently proposed selection-channel-aware JPEG rich model SCA-GFR [18]. For all of the three steganographic algorithms, the performance of Xu's model was unsatisfactory. The degraded performance of Xu's model is acceptable, since it is designed for spatial-domain steganalysis. The superiority of our proposed framework is more obvious in basic500K. This is due to the fact that with more training samples raised by one magnitude, the large-scale basic500K dataset with 500,000 training samples (covers plus the corresponding stegos) is more favor of deep-learning frameworks like the one proposed by us. If only consider the performance of a single model, our proposed framework with *Type1* subnets behaved better than its companion with *Type2* subnets. Furthermore, the final prediction conducted by the ensemble of five independently trained models shows that model ensemble could improve the detection accuracy by 1% regardless of the type of the underlying subnet configurations.[8] Since the performance of our proposed framework with *Type1* subnets is always better than that with *Type2* subnets, we insisted on using *Type1* subnets in the following experiments. However, please note that *Type2* subnet can potentially be used in more complex deep-learning steganalytic frameworks in the future since it is a memory-efficient model.

In Fig. 4 we show how the testing accuracy changes with successive training iterations in the experiments which were conducted on basic50K, basic500K and basic5000K, our largest-scale dataset, respectively. The tests were performed on standalone testing dataset every $10,000$ training iterations and the models were trained for 20×10^4 iterations in total. Only stego images with 0.4bpnzAC were included in the experiments due to the limited computational capacity. Even so for basic5000K there were five million images (covers plus the corresponding stegos) involved in a training epoch. Our proposed deep-learning framework showed strong learning capacity that further improves along with the growth of training samples. From Fig. 4 we can also see that the curve of testing accuracy for the framework trained on basic5000K not only is of the best performance but also is of the best stability. Please note that 20×10^4 iterations is roughly equivalent to 256 epochs for basic50K, 25.6 epochs for bsic500K, and only 2.56 epochs for basic5000K. Therefore the full potential of our proposed framework with large-scale training datasets may not

[7]The original Xu's model is fed with 512×512 images. In order to make it adapt to 256×256 inputs used in our experiments, we explicitly set "stride=2" for its bottom convolutional layer which takes the residual map generated by the KV kernel as input. Please note that we also set "stride=2" in the bottom convolutional layer of *Type1* and *Type2* subnets of our proposed framework.

[8]Please note that the ensemble approach of Xu's model [24] can also probably obtain better results. The experimental results of ensemble prediction of Xu's model are omitted in Fig. 3 for clarity.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2779446, IEEE Transactions on Information Forensics and Security

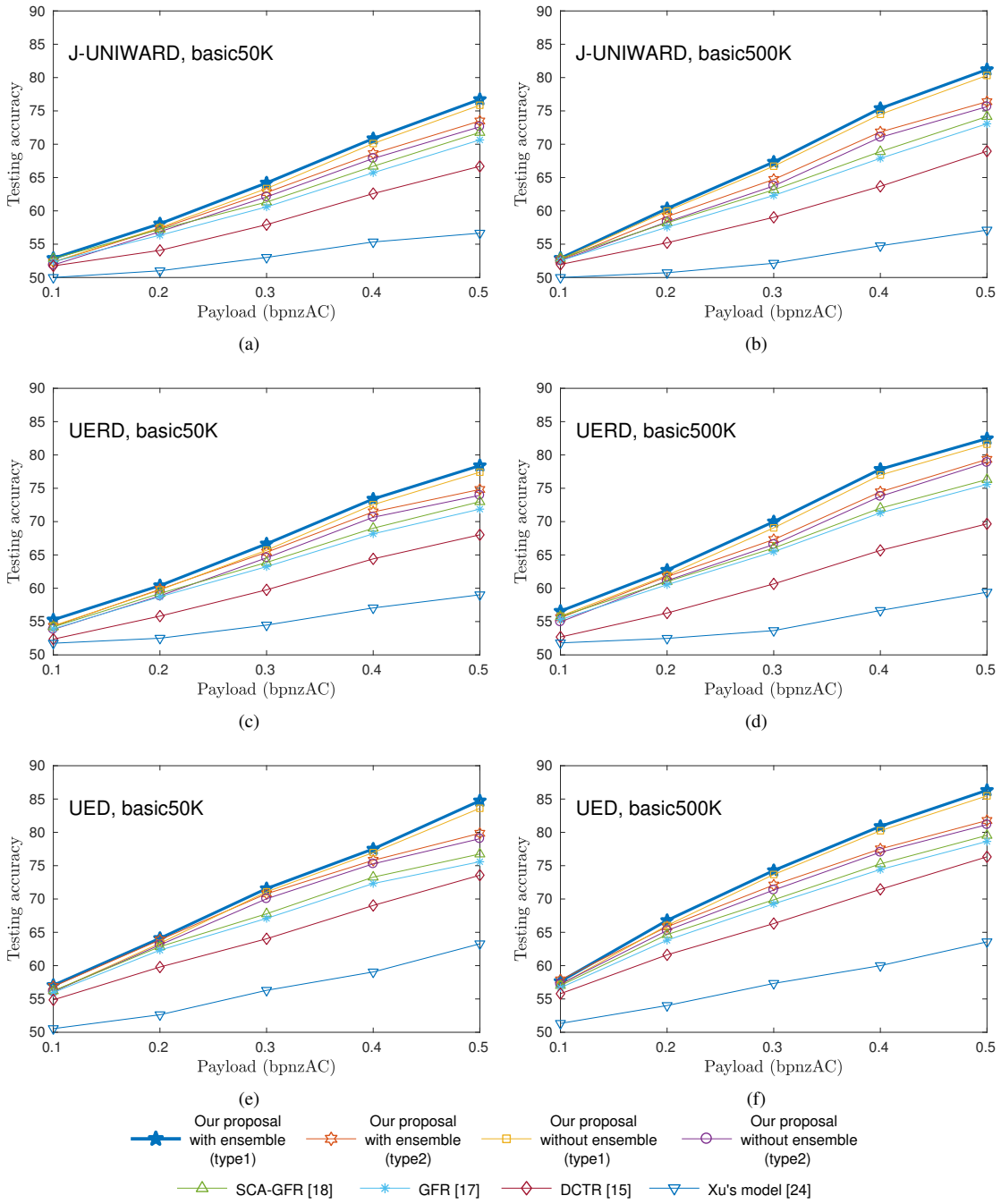


Fig. 3. Comparison of testing accuracy of our proposed frameworks with four steganalytic models described in the literature, two hand-crafted JPEG domain rich models (DCTR and GFR), a selection-channel-aware variant of GFR (SCA-GFR) and a deep-learning steganalytic model proposed by Xu et al. [24]. (a) and (b) are the results for J-UNIWARD; (c) and (d) are for UERD; (e) and (f) are for UED. The experiments for (a), (c) and (e) were conducted on basic50K, while those for (b), (d) and (f) were conducted on basic500K.

have been fully exploited.[9]

Throughout the experiments, our proposed framework ran steadily. During the training procedure, it could accomplish 1,000 iterations every 20 minutes. That is to say, 20×10^4 training iterations could be finished in about 67 hours. With K80 GPU cards, We can expect to finish one epoch of training in 0.26 hour, 2.6 hours, and 26 hours for basic50K, basic500K, and basic5000K, respectively.

[9]The implementation of ensemble classifier [14] used by rich models cannot be scaled to large-scale datasets. Therefore we cannot provide the testing accuracy of DCTR and GFR in basic5000K for comparison in Fig. 4.

D. Performance with mismatched targets, altered blocking artifact, doubled-sized inputs and single-compressed images

First of all, please note that in the following experiments, our proposed framework is equipped with *Type1* subnet. No ensemble prediction is involved to reduce the time of experiments.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2779446, IEEE Transactions on Information Forensics and Security

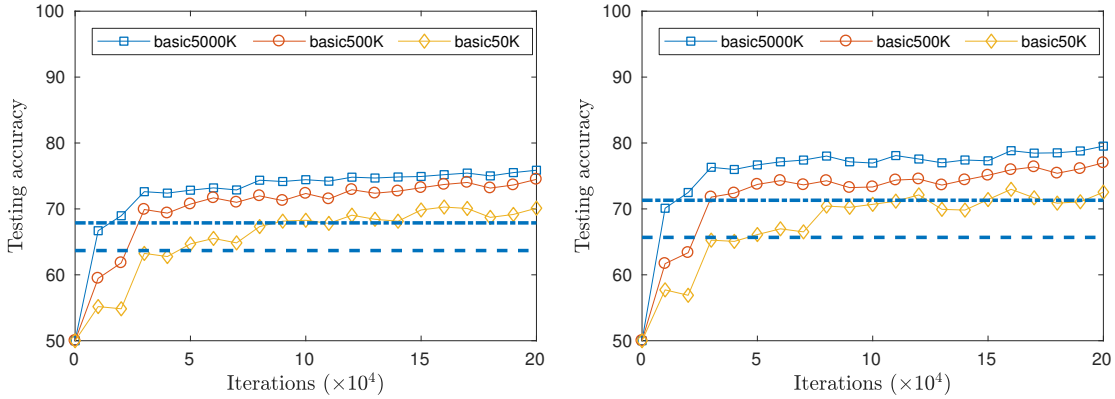


Fig. 4. Testing accuracies versus training iterations for our proposed framework. The experimental results on basic50K, basic500K and basic5000K are reported. For brevity, only stego images with 0.4bpnzAC were included in the experiments. (a) is for J-UNIWARD steganography while (b) is for UERD steganography. In (a) and (b), The dash-dotted and the dashed reference lines denote the best testing accuracy of GFR and DCTR in basic500K, respectively.

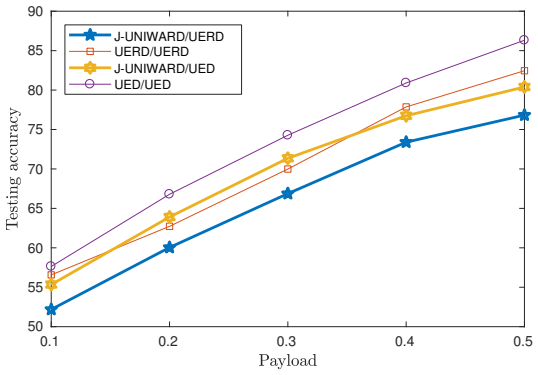


Fig. 5. Comparison of attacking-target transfer ability of our proposed framework. The experiments were conducted on basic500K dataset. Only stego images with 0.4bpnzAC were included in the experiments. The notations in the legend take the form of the target in training and the target in testing delimited by a slash (/). For example, "J-UNIWARD/UERD" means that J-UNIWARD stego images were used in training while UERD stego images were used in testing.

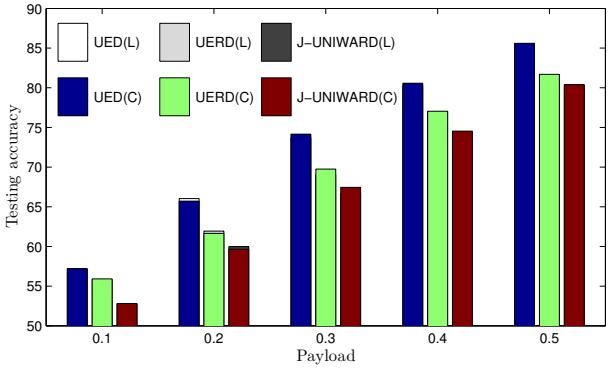


Fig. 6. The impact of altered blocking artifact on the performance of our proposed framework. Only stego images with 0.4bpnzAC were included in the experiments. All of the models were trained on basic500K training set and then tested on the corresponding testing set with central-cropped images. The legend "C" in parentheses denotes those tested on central-cropped images, while "L" in parentheses denotes those tested on the original basic500K testing set. For example, "J-UNIWARD (C)" means that the corresponding framework was trained and tested with J-UNIWARD stego images. It was trained on basic500K training set and then tested on the corresponding testing set with central-cropped images.

In Fig. 5, we observe the attacking-target transfer ability of our proposed framework. The framework was trained with J-UNIWARD cover/stego pairs and then tested with UERD/UED cover/stego pairs. The detection accuracy is roughly $3\% - 4\%$ worse compared with that trained and tested with the same type of stego images. However, the degradation of detection performance is acceptable especially for the detection of UED stego images given that UED works in a very different way compared with J-UNIWARD.

8×8 block processing during JPEG compression introduces blocking artifacts, which can be used as intrinsic statistical characteristic of JPEG cover images. Secret bits embedded in the DCT domain tend to impair blocking artifacts, therefore leave traces which can be utilized by steganalyzers. An interesting problem is to access the performance of our proposed framework depending on the intrinsic statistical characteristic of blocking artifacts. In Fig. 6, we observe the impact of altered blocking artifacts on the performance of our proposed framework. The default testing set in basic500K contains left-top cropped images in which the original DCT grid alignment is preserved. In this experiment for all the testing images in basic500K, we re-compressed their corresponding original images in ImageNet with quality factor 75 and then converted them to grayscale images again. We cropped their central 256×256 regions to constitute a new testing set. The motivation is that central cropping cannot preserve the original DCT grid alignment in most cases, therefore the blocking artifacts from two different sources coexist. As a result the blocking artifacts in the images of the new testing set are different from those in the training set. However, Fig. 6 reveals that the impact of altered blocking artifact on the performance of our proposed framework is small. Our proposed framework has captured more complex intrinsic statistical characteristic besides blocking artifact.

All of the above experiments used images of size 256×256 pixels. This limitation stems mainly from the following two

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2779446, IEEE Transactions on Information Forensics and Security

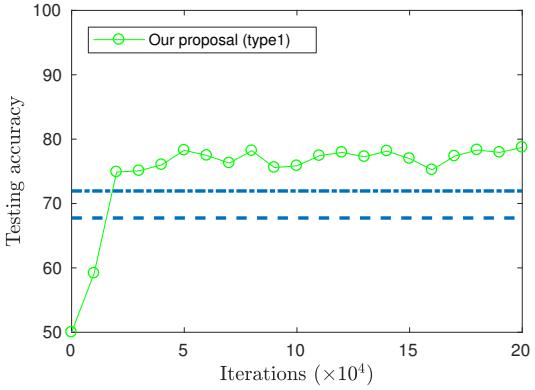


Fig. 7. Testing accuracies versus training iterations for our modified framework which takes 512×512 images as input. Only J-UNIWARD stego images with 0.4bpnzAC are included in the experiment. As in Fig. 4, The dash-dotted reference line denotes the best testing accuracy of GFR, while the dashed reference line denotes the best testing accuracy of DCTR in the same testing dataset.

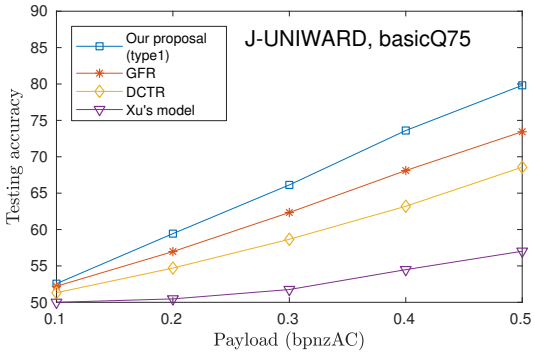


Fig. 8. Comparison of testing accuracy of our proposed framework with GFR, DCTR, and Xu's model for J-UNIWARD on basicQ75 dataset.

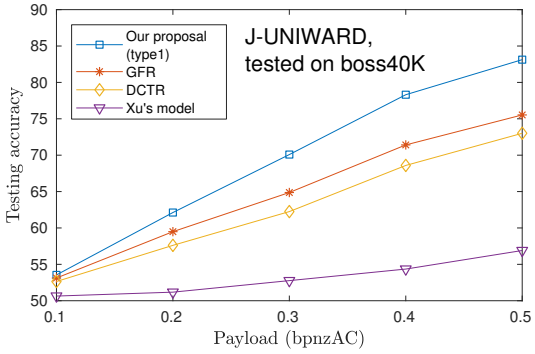


Fig. 9. Comparison of testing accuracy of our proposed framework with GFR, DCTR, and Xu's model for J-UNIWARD on boss40K dataset.

factors: Firstly, target images with larger size, e.g. 512×512 pixels result in deep-learning models hard to train with K80 GPU cards we have in hands. Secondly, large-sized ImageNet images are in the minority. Out of fourteen million ImageNet images, only roughly 0.7 million of them are larger than 512×512 pixels. In the following experiment, we tested our proposed framework with double-sized inputs on this limited dataset. 500 thousand JPEG images with size larger than 512×512 were randomly picked out from ImageNet

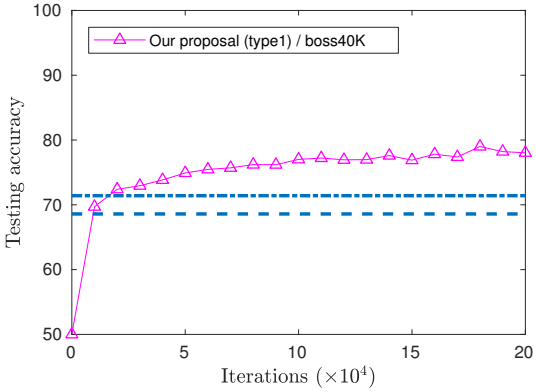


Fig. 10. Testing accuracies versus training iterations for our proposed framework. The models are trained on basic500K while tested on boss40K. Only J-UNIWARD stego images with 0.4bpnzAC are included in the experiment. The dash-dotted line and the dashed line denote the best testing accuracy of GFR and DCTR, respectively.

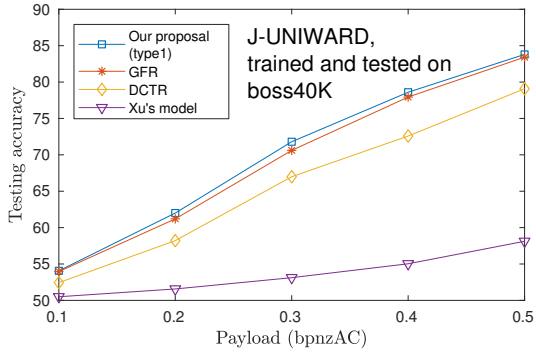


Fig. 11. Comparison of testing accuracy of our proposed framework with GFR, DCTR, and Xu's model for J-UNIWARD on boss40K dataset, when all of them were also trained on boss40K.

and were converted to 512×512 with the same processing procedure as mentioned in Sect. III-A. Due to GPU memory constraints, we simplified the model by using doubled stride in the convolutional layer of each subnet (i.e. 4 instead of 2). All other experiment setups were remained the same except that the batch size in the training procedure is reduced to 32. Only J-UNIWARD stego images with 0.4bpnzAC were included in the experiment. Fig. 7 shows the testing accuracy in successive training iterations. The training procedure also converged quickly and delivered better performance than the DCTR and GFR models. Due to the limited computational capacity, subnets with wider and deeper structures were not evaluated in this experiment. Its potential for target images with larger size may have not been fully demonstrated.

Up to now we used double-compressed images in the experiments. As reported by Pibre et al. [23], CNN based steganalyzers can take advantage of seemingly irrelevant subtle patterns to boost their performance. We must eliminate the possibility that our proposed framework makes use of the double compression artifacts to dispel the doubts of the colleagues. Hence, we conducted two more experiments with single-compressed JPEG images.

Firstly, There are about $410,000$ ImageNet images can be confirmed as being compressed with quality factor 75.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2779446, IEEE Transactions on Information Forensics and Security

They were all selected. Their left-top 256×256 regions were cropped, converted to grayscale without double compression to constitute a new dataset "basicQ75". 200,000 cover images were randomly selected from them for training while the rest were for testing. In Fig. 8, we compare the performance of our proposed framework with three other steganalyzers for J-UNIWARD on bacicQ75 dataset. For the sake of brevity, only the results of half of the steganalyzers listed in Fig. 3 are listed in Fig. 8. However, by comparing Fig. 8 and Fig. 3(b), we still can find that as with all other three steganalyzers, our proposed framework only suffered slight performance degradation, which may be attributed to the relative lack of diversity in basicQ75 dataset.

Secondly, we divided every image in BOSSBase public dataset [28] into four equal parts and then JPEG compressed them with quality factor 75. Through this method, we obtained 40,000 single-compressed JPEG cover images. We denoted them as "boss40K" dataset, and used all of the 40,000 cover images and the corresponding stego images to test the performance of our proposed framework and other steganalyzers trained on basic500K. We prefer to use all of the images in boss40K dataset in testing rather than in training, which is based on the following two aspects: 1. Merely 40,000 images are not suitable for training a deep-learning steganalyzer with hundreds of thousands of learnable parameters. 2. As a dataset with totally different source, boss40K is more suitable for checking transfer ability of steganalyzers trained with ImageNet images.

In Fig. 9, we show the testing results of our proposed framework (with *Type1* subnet, without ensemble) and other steganalyzers on boss40K. Please note that all the steganalyzers used in this experiment were trained on basic500K. By comparing Fig. 9 and Fig. 3(b), we are delighted to find that our proposed framework even achieved better detection performance, and its superiority over all other three steganalyzers became more obvious. Fig. 10 shows testing accuracy of our proposed framework on boss40K in successive training iterations. Please note that the model was also trained on basic500K. From Fig. 10 we can see our proposed framework trained on basic500K exhibited rapid convergence even when evaluated on a dataset with totally different source, which provides complementary evidence to support the removal of validation set in our large-scale experiments.

For the sake of completeness, we also show the testing results of our proposed framework (with *Type1* subnet, without ensemble) and other steganalyzers on boss40K dataset, when all of them were also trained on boss40K in Fig. 11. Since validation cannot be omitted for a small-scale dataset, boss40K was split into 60/15/25 ratio, for training, validating, and testing, respectively. We guaranteed that all the sub-images of a given BOSSBase image could only be assigned to one sub-dataset. Please note that our proposed framework aims at large-scale JPEG image steganalysis. It needs to be fed with a great deal of labeled samples in the training procedure. Therefore from Fig. 11, it is no doubt that superiority of our proposed framework in such a small-scale dataset was not obvious. However, it still retained equal or even slightly better performance than GFR.

TABLE II

COMPARISON OF NUMBER OF PARAMETERS AND COMPUTATIONAL COMPLEXITY FOR OUR PROPOSED FRAMEWORK AND XU'S NEW MODEL. THE COMPUTATIONAL COMPLEXITY IS MEASURED IN TERMS OF FLOPS (FLOATING-POINT OPERATIONS).

	ours with *Type1* subnet	Xu's new model
Parameters	1.66×10^6	4.86×10^6
FLOPs	2.77×10^8	1.53×10^9

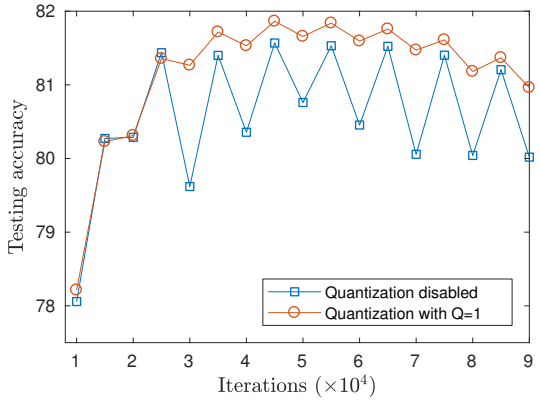


Fig. 12. Testing accuracies versus training iterations for Xu's new model (with or without quantization $Q = 1$ enabled). The experiment was conducted on basic500K. Only J-UNIWARD stego images with 0.4bpnzAC were included in the experiment. We adopted the training settings in Xu's work so that the training of the models were stopped after 9×10^4 iterations. Polyak averaging was enabled, as suggested by Xu.

E. Comparison to newly emerging works

During the course of the review process, we noticed that two new research works in the field of deep-learning JPEG steganalysis have been published [36], [37]. Due to the limited computational capacity, we only conducted a comparative study of our proposed framework and the framework proposed in [36] (referred as Xu's new model) since it also aims at large-scale JPEG image steganalysis.

In [36], Xu compared his framework with this work pre-printed on arXiv, and claimed that his framework can achieve significant performance improvement compared to the implementation of our generic framework illustrated in Fig. 1. However, Please note that as shown in Tab. II, Xu's new model [36] is a behemoth with about triple parameters and more than five times of computational complexity compared with our proposed framework with *Type1* subnet. Therefore it is natural for Xu's new model [36] to achieve better detection performance with multiple expansions in capacity.

Xu deprecated the use of quantization in deep-learning based steganalyzer, which we do not agree with. We conducted a verification experiment. As shown in Fig. 12, on the standalone basic500K testing set which contains 500,000 cover-stego pairs, simply adding back quantization with $Q = 1$ to Xu's new model [36] could not only make the detection performance more stable but also improve testing accuracy. That is to say, even with Xu's new model [36], experimental evidence also supports the introduction of Q&T phase in deep-learning steganalyzers, and supports our opinion that recognizing threshold quantizers as a whole.

As mentioned in Sect. II-C, what proposed in this pa-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2779446, IEEE Transactions on Information Forensics and Security

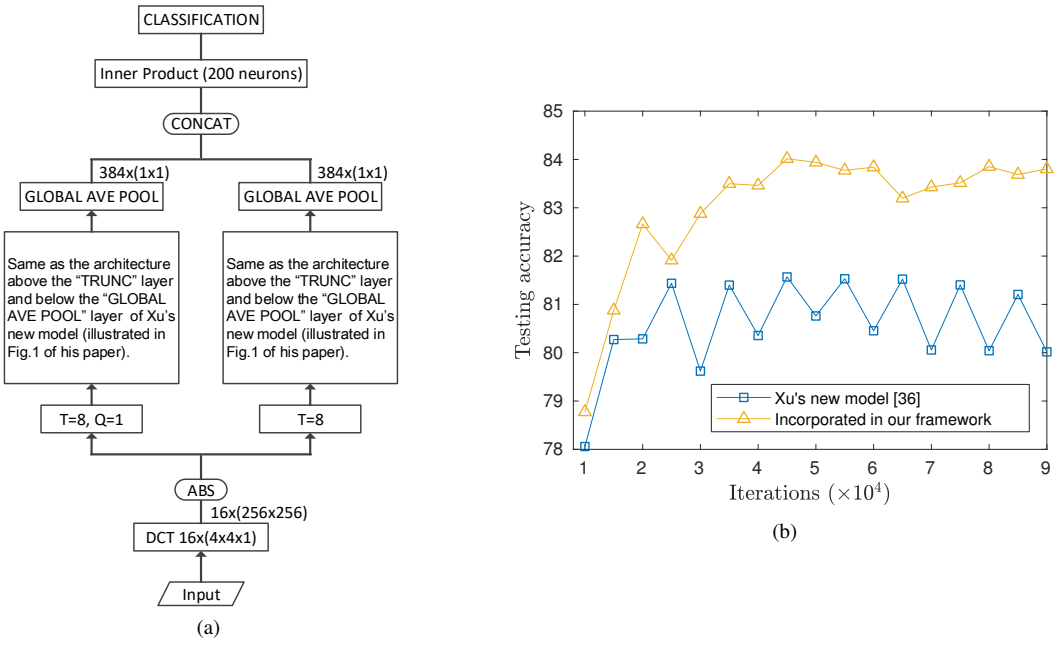


Fig. 13. (a) Conceptual architecture of our proposed hybrid deep-learning framework incorporated with Xu's new model [36]. (b) Testing accuracies versus training iterations for Xu's new model [36] and our hybrid framework incorporated with Xu's new model as shown in (a). The experimental setup is the same as in Fig. 12.

per is a generic hybrid architecture for deep-learning JPEG steganalyzers. It is composed of two stages. The first stage is equipped with hand-crafted model parameters, while the second stage is a compound deep CNN network with a sequence of independent subnets, and the actual number of the subnets is determined by the Q&T combinations experimentally. Newly emerging deep-learning steganalyzers can be used as the prototypes of the subnets in the second stage of our proposed hybrid architecture. Via this way, we can incorporate them into our framework. For example, *Type1* subnet used in our work is inspired by Xu's model [24]. Certainly we can also incorporate Xu's new model [36] into our framework. However, a complete incorporation of Xu's new model [36] in our framework involves a great deal of experiments for architecture adjustments (e.g. evaluating different Q&T combinations), and is beyond the scope of this work. Here we just provide a straightforward incorporation to demonstrate the generality and potentiality of our proposed framework. As shown in Fig. 13(a), Xu's new model [36] is incorporated in our hybrid framework as the prototype of two subnets, one is with ($T = 8, Q = 1$) while another is with ($T = 8$) and quantization disabled (the original setting in Xu's new model [36]). Fig. 13(b) shows the testing accuracy in successive training iterations for this new hybrid framework and Xu's new model [36]. From Fig. 13(b), we can see our proposed framework incorporated with Xu's new model outperformed the original one by a clear margin. We expect that greater performance improvement can be achieved with more complete incorporation of Xu's new model [36] in our hybrid generic framework.

IV. Concluding remarks

Application of deep-learning frameworks in image steganalysis has drawn attention of many researchers. In this paper we proposed a hybrid deep-learning framework for large-scale JPEG steganalysis, which for the first time utilize quantization and truncation into deep-learning steganalyzers. We have provided experimental and theoretical testimonies to support the utilization of quantization and truncation in the proposed framework. Our proposed framework is generic, so that existing deep-learning based steganalyzers is easy to be incorporated into it as a subnet prototype. We have demonstrated the capacity of the proposed framework with different subnet configurations, including one that incorporated from a new JPEG deep-learning steganalyzer emerged during the review process. The extensive experiments conducted on a large-scale dataset extracted from ImageNet clearly show that our proposed framework provides a boost of performances with quantitative metrics.

Our future work will focus on two aspects: (1) incorporation of adversarial machine learning into our proposed framework to make it jointly optimized with its opponent; (2) further exploration of the application of our proposed framework in the field of multimedia forensics.

Appendix A
Theoretical reflection

State-of-the-art steganalytic feature extractors, either in spatial domain or in JPEG domain, take the spatial representation (usually type-casted to real) of target image as input [11]–[13], [15]–[18], [20]–[25], [27]. Furthermore, please note that JPEG steganalytic feature extractors are usually fed with decompressed (non-rounded and non-truncated) JPEG images. We follow this approach in our research. Therefore, a grayscale

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2779446, IEEE Transactions on Information Forensics and Security

input image can be represented as $\mathbf{X} = (x_{pq})^{M\times N} = \mathbf{C} + \mathbf{N}$, where $\mathbf{C} = (c_{pq})^{M\times N}$, $c_{pq} \in \mathbb{R}$ denotes the corresponding cover image, and $\mathbf{N} = (n_{pq})^{M\times N}$, $n_{pq} \in \mathbb{R}$ denotes the additive stego noise [10].

Our reflection starts from one easily-verified fact: the magnitude of most of the elements of $\mathbf{N}$ matrix remain tiny with respect to the corresponding elements of $\mathbf{C}$ even for a stego image with high embedding rate (on average close to two orders of magnitude larger). State-of-the-art content-adaptive steganography, whether in spatial domain or in JPEG domain, tends to embed secret bits in highly textured area. As a result, even filtered by state-of-the-art steganalytic kernels (e.g. KV kernel used in [21], [23], [24]) the magnitudes of most of the filtered residual elements are still much larger than the corresponding stego noises.

Suppose that we apply a convolutional layer with kernels of size of $m \times n$, and suppose we take as input $\mathbf{X}$ of size $M \times N$. Since in the context the input and output of a convolutional layer are of two-dimensions, we adopt two-dimensional indexing here. Convolution is just a dot product with local-connected-and-shared weights. That is to say, for each given $z^{(2)}_{rs}$, it is only the weighted sum of lower-layer inputs located in a $m \times n$ local area with index (r, s) as its centre irrespective of boundary condition, and the weights used in the weighted sum are shared in the calculation of all the $z^{(2)}_{rs}$, $1 \le r \le M$, $1 \le s \le N$.

By rewriting (1) using two-dimensional indexing, setting $l = 1$, $a^{(1)}_{pq} = x_{pq}$ and restrict the size of the dot product to $m\times n$ (m and n assume to be odd to omit unimportant details), we get:

$$\begin{aligned} z^{(2)}_{rs} &= \sum_{p=1}^{M}\sum_{q=1}^{N} W^{(1)}_{pq,rs}x_{pq} + b^{(1)}_{rs} \\ &= \sum_{p=1}^{m}\sum_{q=1}^{n} W^{(1)}_{(r-\lceil\frac{m}{2}\rceil+p)(s-\lceil\frac{n}{2}\rceil+q),rs}c_{(r-\lceil\frac{m}{2}\rceil+p)(s-\lceil\frac{n}{2}\rceil+q)} + \\ &\quad \sum_{p=1}^{m}\sum_{q=1}^{n} W^{(1)}_{(r-\lceil\frac{m}{2}\rceil+p)(s-\lceil\frac{n}{2}\rceil+q),rs}n_{(r-\lceil\frac{m}{2}\rceil+p)(s-\lceil\frac{n}{2}\rceil+q)} + b^{(1)}_{rs} \end{aligned} \quad (10)$$

In (10), $\lceil\cdot\rceil$ denotes the ceiling operation. From (10) we can see that if the convolutional layer is initialized with kernels which are already sensitive to the stego noise (e.g. KV kernel) or is regularized as high-pass as proposed in [34], then $\sum_{p=1}^{m}\sum_{q=1}^{n} W^{(1)}_{(r-\lceil\frac{m}{2}\rceil+p)(s-\lceil\frac{n}{2}\rceil+q),rs}c_{(r-\lceil\frac{m}{2}\rceil+p)(s-\lceil\frac{n}{2}\rceil+q)}$ can be suppressed. However, as we mentioned above, the magnitudes of most of the filtered residual elements are still much larger than the corresponding stego noises, and the accumulation in (10) helps reduce the influence of outliers. Therefore in either scenario, on average $\sum_{p=1}^{m}\sum_{q=1}^{n} W^{(1)}_{(r-\lceil\frac{m}{2}\rceil+p)(s-\lceil\frac{n}{2}\rceil+q),rs}c_{(r-\lceil\frac{m}{2}\rceil+p)(s-\lceil\frac{n}{2}\rceil+q)}$ still accounts for the vast majority magnitude when compared with $\sum_{p=1}^{m}\sum_{q=1}^{n} W^{(1)}_{(r-\lceil\frac{m}{2}\rceil+p)(s-\lceil\frac{n}{2}\rceil+q),rs}n_{(r-\lceil\frac{m}{2}\rceil+p)(s-\lceil\frac{n}{2}\rceil+q)}$ in (10).

[10]For JPEG steganography, the additive stego noise is directly added to quantized DCT coefficients. However, the linearity property of the DCT/IDCT transform guarantees that the corresponding stego noise in the spatial-domain representation is still additive.

For a given index $(\hat{p}, \hat{q})$ where $\hat{p} = r-\lceil\frac{m}{2}\rceil+p$, $\hat{q} = s-\lceil\frac{n}{2}\rceil+q$, according to (4) and (5) we can see that when the gradient is backpropagated to the layer L_1:

$$\frac{\partial}{\partial W^{(1)}_{\hat{p}\hat{q},rs}} J(W, b; x^{(h)}, y^{(h)}) = x_{\hat{p}\hat{q}} \cdot \vartheta^{(2)}_{rs} = (c_{\hat{p}\hat{q}} + n_{\hat{p}\hat{q}}) \cdot \vartheta^{(2)}_{rs} \quad (11)$$

in which:

$$\vartheta^{(2)}_{rs} = (\sum_{k} W^{(2)}_{rs,k}\vartheta^{(3)}_{k}) f'(z^{(2)}_{rs}) \quad (12)$$

In (12) $\sum_k W^{(2)}_{rs,k}\vartheta^{(3)}_k$ is fixed when the gradient is backpropagated to the layer L_2. As a result $\vartheta^{(2)}_{rs} \propto f'(z^{(2)}_{rs})$. Please note that $f'(z^{(2)}_{rs})$ is the derivative of the activation function of $z^{(2)}_{rs}$. The derivatives of all of the existing practical activation functions, including Sigmoid, TanH, and ReLU, have narrow ranges. And furthermore, if only consider the curve in positive axis (or negative axis), it is easy to verify that they are linear, or quasi-linear, namely:

$$\min\{f'(z_1), f'(z_2)\} \le f'(\lambda z_1 + (1-\lambda)z_2) \le \max\{f'(z_1), f'(z_2)\}, \quad (13)$$

for any $\lambda \in (0, 1)$ and $z_1 \ne z_2$. Based on the fact that $\vartheta^{(2)}_{rs} \propto f'(z^{(2)}_{rs})$, $\vartheta^{(2)}_{rs}$ is proportional/inverse proportional to, or quasi-proportional/inverse quasi-proportional to $z^{(2)}_{rs}$ provided the polarity of $z^{(2)}_{rs}$ remains the same. Return to (10). Since $\sum_{p=1}^{m}\sum_{q=1}^{n} W^{(1)}_{(r-\lceil\frac{m}{2}\rceil+p)(s-\lceil\frac{n}{2}\rceil+q),rs}c_{(r-\lceil\frac{m}{2}\rceil+p)(s-\lceil\frac{n}{2}\rceil+q)}$ accounts for the vast majority magnitude, with or without $\sum_{p=1}^{m}\sum_{q=1}^{n} W^{(1)}_{(r-\lceil\frac{m}{2}\rceil+p)(s-\lceil\frac{n}{2}\rceil+q),rs}n_{(r-\lceil\frac{m}{2}\rceil+p)(s-\lceil\frac{n}{2}\rceil+q)}$, the polarity of $z^{(2)}_{rs}$ will not change. Therefore the linearity (quasi-linearity) between $\vartheta^{(2)}_{rs}$ and $z^{(2)}_{rs}$ holds. Consequently, due to the linearity (quasi-linearity) between $\vartheta^{(2)}_{rs}$ and $z^{(2)}_{rs}$, the magnitude of $\vartheta^{(2)}_{rs}$ mainly depends on the weighted sum of the cover image pixels located in the corresponding $m \times n$ local area, rather than the weighted sum of those stego noises.

Furthermore, in (11) we can see there is a multiply factor to $\vartheta^{(2)}_{rs}$, $(c_{\hat{p}\hat{q}} + n_{\hat{p}\hat{q}})$. Since by average $|c_{\hat{p}\hat{q}}|$ is close to two orders of magnitude larger than $|n_{\hat{p}\hat{q}}|$ even with a high embedding rate, the impact of the neighboring cover image pixels on $\frac{\partial}{\partial W^{(1)}_{\hat{p}\hat{q},rs}} J(W, b; x^{(h)}, y^{(h)})$ is further amplified. As a result, the influence of $n_{\hat{p}\hat{q}}$, and the neighboring stego noise in the corresponding $m \times n$ local area, to $\frac{\partial}{\partial W^{(1)}_{\hat{p}\hat{q},rs}} J(W, b; x^{(h)}, y^{(h)})$ becomes very weak. At last, since in a convolutional layer the weights are shared, all the partial derivatives with respect to a given shared weight should be accumulated:

$$\frac{\partial}{\partial W^{(1)}_{pq}} J(W, b; x^{(h)}, y^{(h)}) = \sum_{r=1}^{M}\sum_{s=1}^{N} \frac{\partial J(W, b; x^{(h)}, y^{(h)})}{\partial W^{(1)}_{(r-\lceil\frac{m}{2}\rceil+p)(s-\lceil\frac{n}{2}\rceil+q),rs}}, \quad 1 \le p \le m,\ 1 \le q \le n. \quad (14)$$

The accumulation in (14) again helps reduce the influence of outliers. As a result, it is safe for us to make a conclusion that the influence of stego noises to $\frac{\partial}{\partial W^{(1)}_{pq}} J(W, b; x^{(h)}, y^{(h)})$, $1 \le p \le m,\ 1 \le q \le n$ is weak in statistical sense. Consequently, gradient descent algorithm in the bottom convolutional layer will be always guided by the cover image contents rather than the stego noises. In other words, the optimization of the bottom convolutional layer in favor of the extraction of stego noises is hard to achieve with gradient descent.

Acknowledgment

The authors would like to thank DDE Laboratory in SUNY Binghamton and Dr. Guanshuo Xu for sharing the source code of their steganalysis models online. We also appreciate Prof. Jiangqun Ni in Sun Yat-sen University, China for permission to use their implementation of UED and UERD in our experiments. Specifically, we are grateful to Dr. Paweł Korus at that time in Shenzhen University for valuable advice.

References

[1] J. Fridrich and T. Filler, "Practical methods for minimizing embedding impact in steganography," in *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, 2007, pp. 650 502–1–650 502–15.

[2] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proc. 12th Information Hiding Workshop (IH'2010)*, 2010, pp. 161–177.

[3] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Proc. IEEE 2014 International Conference on Image Processing, (ICIP'2014)*, 2014, pp. 4206–4210.

[4] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 221–234, 2016.

[5] L. Guo, J. Ni, and Y. Q. Shi, "Uniform embedding for efficient JPEG steganography," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 5, pp. 814–825, 2014.

[6] L. Guo, J. Ni, W. Su, C. Tang, and Y. Q. Shi, "Using statistical image model for JPEG steganography: Uniform embedding revisited," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2669–2680, 2015.

[7] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, pp. 1–13, 2014.

[8] T. Denemark and J. Fridrich, "Improving steganographic security by synchronizing the selection channel," in *Proc. 3rd ACM Information Hiding and Multimedia Security Workshop (IH&MMSec'2015)*, 2015, pp. 5–14.

[9] B. Li, M. Wang, X. Li, S. Tan, and J. Huang, "A strategy of clustering modification directions in spatial image steganography," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1905–1917, 2015.

[10] T. Denemark and J. Fridrich, "Side-informed steganography with additive distortion," in *Proc. 7th IEEE International Workshop on Information Forensic and Security (WIFS'2015)*, 2015, pp. 1–6.

[11] J. Fridrich and J. Kodovský, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[12] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich, "Selection-channel-aware rich model for steganalysis of digital images," in *Proc. 6th IEEE International Workshop on Information Forensic and Security (WIFS'2014)*, 2014, pp. 48–53.

[13] W. Tang, H. Li, W. Luo, and J. Huang, "Adaptive steganalysis based on embedding probabilities of pixels," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 734–745, 2016.

[14] J. Kodovský and J. Fridrich, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2012.

[15] V. Holub and J. Fridrich, "Low-complexity features for JPEG steganalysis using undecimated DCT," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 219–228, 2015.

[16] ——, "Phase-aware projection model for steganalysis of JPEG images," in *Proc. IS&T/SPIE Electronic Imaging 2015 (Media Watermarking, Security, and Forensics)*, 2015, pp. 94 090T–1–94 090T–11.

[17] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang, "Steganalysis of adaptive JPEG steganography using 2d Gabor filters," in *Proc. 3rd ACM Information Hiding and Multimedia Security Workshop (IH&MMSec'2015)*, 2015, pp. 15–23.

[18] T. Denemark, M. Boroumand, and J. Fridrich, "Steganalysis features for content-adaptive JPEG steganography," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1736–1746, 2016.

[19] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[20] S. Tan and B. Li, "Stacked convolutional auto-encoders for steganalysis of digital images," in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA'2014)*, 2014.

[21] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," in *Proc. IS&T/SPIE Electronic Imaging 2015 (Media Watermarking, Security, and Forensics)*, 2015, pp. 94 090J–1–94 090J–10.

[22] ——, "Learning and transferring representations for image steganalysis using convolutional neural network," in *Proc. IEEE 2016 International Conference on Image Processing, (ICIP'2016)*, 2016, pp. 2752–2756.

[23] L. Pibre, P. Jérôme, D. Ienco, and M. Chaumont, "Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover source-mismatch," in *Proc. Media Watermarking, Security, and Forensics, Part of IS&T International Symposium on Electronic Imaging (EI'2016)*, San Francisco, CA, USA, 14-18 February 2016.

[24] G. Xu, H. Z. Wu, and Y. Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708–712, 2016.

[25] ——, "Ensemble of CNNs for steganalysis: An empirical study," in *Proc. 4th ACM Information Hiding and Multimedia Security Workshop (IH&MMSec'2016)*, 2016, pp. 103–107.

[26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv:1502.03167*, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[27] V. Sedighi and J. Fridrich, "Histogram layer, moving convolutional neural networks towards feature-based steganalysis," in *Proc. Media Watermarking, Security, and Forensics, Part of IS&T International Symposium on Electronic Imaging (EI'2017)*, Burlingame, CA, 29 Juanuary-2 February 2017.

[28] P. B. T. Filler and T. Pevný, "Break our steganographic system—the ins and outs of organizing BOSS," in *Proc. 13th Information Hiding Workshop (IH'2011)*, 2011, pp. 59–70.

[29] V. Sedighi, J. Fridrich, and R. Cogranne, "Toss that BOSSbase, Alice!" in *Proc. Media Watermarking, Security, and Forensics, Part of IS&T International Symposium on Electronic Imaging (EI'2016)*, San Francisco, CA, USA, 14-18 February 2016.

[30] J. Zeng, S. Tan, and B. Li, "Pre-training via fitting deep neural network to rich-model features extraction procedure and its effect on deep learning for steganalysis," in *Proc. Media Watermarking, Security, and Forensics, Part of IS&T International Symposium on Electronic Imaging (EI'2017)*, Burlingame, CA, USA, 29 Juanuary- 2 February 2017.

[31] "ImageNet," http://image-net.org/, [Online].

[32] "CS231n: Convolutional Neural Networks for Visual Recognition," http://cs231n.github.io/, [Online].

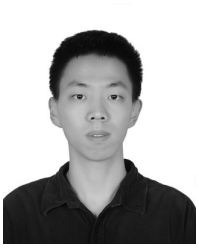
[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[34] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proc. 4th ACM Information Hiding and Multimedia Security Workshop (IH&MMSec'2016)*, 2016, pp. 5–10.

[35] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv:1408.5093*, 2014. [Online]. Available: http://arxiv.org/abs/1408.5093

[36] G. Xu, "Deep convolutional neural network to detect J-UNIWARD," in *Proc. 5th ACM Information Hiding and Multimedia Security Workshop (IH&MMSec'2017)*, 2017, pp. 67–73.

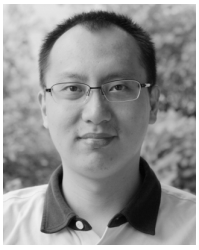
[37] M. Chen, V. Sedighi, M. Boroumand, and J. Fridrich, "JPEG-phase-aware convolutional neural network for steganalysis of JPEG images," in *Proc. 5th ACM Information Hiding and Multimedia Security Workshop (IH&MMSec'2017)*, 2017, pp. 75–84.



Jishen Zeng (S'16) received the B.S degree of electronic information science and technology from Sun Yat-sen University, Guangzhou, China in 2015. He is currently a Ph.D. student in Shenzhen University majoring in information and communication engineering. His current research interests include steganography, steganalysis, multimedia forensics, and deep learning.

Shunquan Tan (M'10–SM'17) received the B.S. degree in computational mathematics and applied software and the Ph.D. degree in computer software and theory from Sun Yat-sen University, Guangzhou, China, in 2002 and 2007, respectively.

He was a Visiting Scholar with New Jersey Institute of Technology, Newark, NJ, USA, from 2005 to 2006. He is currently an Associate Professor with College of Computer Science and Software Engineering, Shenzhen University, China. He is also a member of the Shenzhen Key Laboratory of Media Security. His current research interests include steganography, steganalysis, multimedia forensics, and deep machine learning.



Bin Li (S'07–M'09–SM'17) received the B.E. degree in communication engineering and the Ph.D. degree in communication and information system from Sun Yat-sen University, Guangzhou, China, in 2004 and 2009, respectively.

He was a Visiting Scholar with New Jersey Institute of Technology, Newark, NJ, USA, from 2007 to 2008. He is currently an Associate Professor with Shenzhen University, Shenzhen, China, where he joined in 2009. He is also a member of Shenzhen Key Laboratory of Media Security. His current research interests include image processing, multimedia forensics, and pattern recognition.

Jiwu Huang (M'98–SM'00–F'16) received the B.S. degree from Xidian University, Xian, China, in 1982, the M.S. degree from Tsinghua University, Beijing, China, in 1987, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1998. He is currently a Professor with the College of Information Engineering, Shenzhen University, Shenzhen, China. His current research interests include multimedia forensics and security. He served as an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY from 2010 to 2014 and a member of the Information Forensics and Security Technical Committee, IEEE Signal Processing Society, from 2012 to 2013. He was a General Co-Chair of the IEEE Workshop on Information Forensics and Security in 2013. He is a fellow of IEEE.